



Avis de Soutenance

Monsieur Ke DU

Présentera ses travaux en soutenance

Soutenance prévue le **mardi 10 avril 2018** à 14h00

Lieu : IUT Belfort-Montbéliard, 19 avenue du Maréchal Juin, 90000 Belfort
salle Amphithéâtre 4, Batiment Génie Civil

Titre des travaux : Création et analyse de graphes de traitements sur FPGA, sous contraintes matérielles et contexte temps réel dur

Ecole doctorale : SPIM - Sciences Physiques pour l'Ingénieur et Microtechniques

Section CNU : 27

Unité de recherche : FEMTO-ST Franche Comté Electronique Mécanique Thermique et Optique - Sciences et Technologies

Directeur de thèse : Michel LENCZNER

Codirecteur de thèse : Stéphane DOMAS HDR NON HDR

Soutenance : Publique A huis clos

Membres du jury :

<u>Nom</u>	<u>Qualité</u>	<u>Etablissement</u>	<u>Rôle</u>
M. Michel LENCZNER	Professeur des Universités	Université Bourgogne - Franche-Comté	Directeur de these
M. Eric MONMASSON	Professeur des Universités	Université de Cergy Pontoise	Rapporteur
M. François AUGER	Professeur des Universités	Université de Nantes	Rapporteur
M. Serge WEBER	Professeur des Universités	Université de Lorraine	Examineur
M. Stéphane DOMAS	Maître de Conférences	Université Bourgogne Franche-Comté	CoDirecteur de these

Résumé de la thèse (en français) :

Avec le développement de l'industrie électronique, on constate un nombre croissant de projets avec des contraintes matérielles et temporelles de plus en plus élevées, ce qui conduit à l'utilisation de FPGA (Field Programmable Gate Arrays). Pour cela, le concepteur doit avoir une bonne connaissance de la programmation VHDL car cela nécessite beaucoup de formation et de pratique pour maîtriser ces architectures. Mais même pour les spécialistes, le processus de développement prend beaucoup de temps. Par conséquent, le développement d'un outil pour aider les utilisateurs non experts à travailler sur FPGA est nécessaire. Des outils tels que Simulink+HDL coder proposent une interface graphique pour créer un design en posant des blocs sur un tableau et en les connectant. Malheureusement, ce type d'outil souffre de deux défauts. Le premier est qu'il ne prend pas en compte les caractéristiques physiques de l'architecture cible. L'autre est qu'il ne vérifie pas si les flux de données entrant sont traités correctement par le design. Cela oblige le développeur à créer de nombreux tests, ce qui est fastidieux et consommateur en temps. Par conséquent, ce n'est pas une solution adaptée pour produire des applications dans un environnement en temps réel et des contraintes matérielles strictes. Pour gérer la complexité et la taille croissante des designs, l'abstraction est devenue graduellement essentielle. Des modèles ont émergé afin de représenter un design comme un graphe d'acteurs (c.a.d. de blocs), avec une analyse statique de l'exécution du graphe. Néanmoins, ces modèles sont basés sur une description plus ou moins fidèle du comportement d'architecture réelles telles que les FPGAs. Dans cette thèse, nous nous concentrons sur l'étude d'un nouveau modèle et d'un nouvel outil logiciel pour aider les utilisateurs non experts à concevoir automatiquement des implémentations correctes de FPGA. Les principales contributions sont résumées comme suit: 1. Les limitations des modèles SDF existants, en particulier ceux du modèle SDF-AP, sont décrites et illustrées par l'analyse d'exemples caractéristiques. Les deux problèmes les plus courants rencontrés dans les implémentations d'assemblages de blocs sont la production de résultats incorrects et la croissance infinie de la taille du tampon. 2. Nous proposons un nouveau modèle appelé "Actors with Stretchable Access Patterns" (ASAP) qui décrit le comportement matériel de façon moins limitée que les approches antérieures. Il s'agit d'une manière originale de résoudre le problème d'ordonnancement des acteurs, adaptée aux FPGAs. Il permet de déterminer l'exactitude mathématique d'une exécution sans lancer de simulations complexes. Il peut non seulement modéliser correctement les comportements des acteurs, mais aussi éviter les inconvénients mentionnés ci-dessus. Des algorithmes implémentant ces principes sont également fournis. 3. Nous avons étudié des stratégies et des algorithmes connexes pour analyser un graphe représentant un design. L'exactitude du traitement peut être analysée par une série d'algorithmes permettant par exemple la vérification de la vitesse des flux et la vérification de la compatibilité des patterns. Il est ainsi possible de calculer la vitesse de décimation ou la longueur de délais à appliquer sur les entrées lorsqu'une erreur de correction est détectée. 4. Un logiciel d'aide à la création de design est également développé. Il est appelé BIAST (Block Assembly Tool) et vise à compenser les inconvénients des outils similaires tels que Simulink + HDL. Dans BIAST, les algorithmes du modèle ASAP sont utilisés pour vérifier que pour un flux d'entrée donné, le système peut produire un résultat correct et finalement générer des codes VHDL directement utilisables sur une carte FPGA réelle. De plus, l'outil détermine automatiquement les décimations et les modifications requises. Ainsi, un utilisateur sans aucune compétence de programmation, est capable créer un design pour FPGA.

Abstract (in English)

With the development of electronic industry, a growing number of projects require real-time streaming applications on embedded platforms. These comprise increasingly high hardware and timing constraints, which leads to the use of FPGAs (Field Programmable Gate Arrays). Usually, the designer should have a good knowledge of programming with VHDL or Verilog HDL. Unfortunately, only specialists can do it, because this needs a lot of training and practices to master these architectures. Furthermore, even for specialists, the process of development is quite time consuming. Therefore, how to develop a tool to help non-expert users working on FPGA is a promising but challenging work. Tools like Simulink+HDL coder provide a graphical interface to create a design, by putting functional blocks on a layer and to connect them. Nevertheless, such tools are generally suffering from two flaws. One is that they do not take the physical characteristics of the target architecture of the application into account, including that of the selected FPGA. The other one is that they do not check whether a data stream is processed correctly by the design, besides creating many test-benches, which is tedious and time consuming for the developer. Therefore, they are not suitable to produce applications in real-time environment and high hardware constraints. In order to manage the ever-increasing size and complexity of designs, the abstraction is gradually more and more essential. Some models have emerged to represent a design as a graph of actors (i.e. blocks), with a static analysis of the graph execution. Nevertheless, they have an unfaithful description of the behavior real architectures like an FPGA. In this dissertation, we concentrate on the study of a novel model and software tool that can help non-expert users for automatic design of FPGA implementations correctly. The main contributions are summarized as follows: 1. The limits of existing SDF models, in particular those of the SDF-AP model, are described and illustrated by the analysis of characteristic examples. The two most common problems encountered in block assembly implementations are the production of incorrect results and the infinite growth of buffer size. 2. We propose a new model called Actors Stretchable Access Patterns (ASAP) that describes the hardware behaviors as efficiently and precisely as possible. This is a novel way to address the scheduling problem of actors, adapted to FPGA architectures. It opens the possibility to determine the execution correctness mathematically without launching complex simulations. It can not only model actors' behaviors properly, but also avoid the above mentioned drawbacks. Algorithms that implement these principles also provided. 3. We investigate strategies and related algorithms to analyze a graph representing a designed system. Its correctness can be analyzed by a series of algorithms, such as sample rate checking and pattern compatibility checking. The decimation rate or the delay length to be applied on actor's input can be computed when a correctness failure is detected. This increases the number of possible real FPGA implementations covered by the block assembly method. 4. A software tool based on the concept of functional block graph is also developed. It is called BIAST (Block Assembly Tool) and aims to compensate the drawbacks of other tools based on the same concepts, as for example Simulink + HDL coder. In BIAST, the proposed ASAP model and related algorithms are used to check that for a given input stream, whether the system can produce a correct result and finally generate VHDL code directly usable on a real FPGA-based board. Otherwise, the tool determines the required decimations and modifications on the graph automatically. It makes a user without any programming skills to make designs on FPGAs thanks to the friendly graphic interface.